SOFTWARE TOOL ARTICLE

MetaGenSense: A web application for analysis and visualization of high throughput sequencing metagenomic data [version 1; peer review: 2 approved with reservations, 1 not approved]

Damien Correia¹, Olivia Doppelt-Azeroual²*, Jean-Baptiste Denis³, Mathias Vandenbogaert¹, Valérie Caro¹

¹Pôle Génotypage des Pathogènes, Unité Environnement et Risques Infectieux, Institut Pasteur, F-75724, Paris, France
²Centre d’Informatique pour la Biologie (CIB), Institut Pasteur, F-75724, Paris, France
³Groupe Exploitation et Infrastructure, Institut Pasteur, F-75724, Paris, France

* Equal contributors

Abstract
The detection and characterization of emerging infectious agents has been a continuing public health concern. High Throughput Sequencing (HTS) or Next-Generation Sequencing (NGS) technologies have proven to be promising approaches for efficient and unbiased detection of pathogens in complex biological samples, providing access to comprehensive analyses. As NGS approaches typically yield millions of putatively representative reads per sample, efficient data management and visualization resources have become mandatory. Most usually, those resources are implemented through a dedicated Laboratory Information Management System (LIMS), solely to provide perspective regarding the available information. We developed an easily deployable web-interface, facilitating management and bioinformatics analysis of metagenomics data-samples. It was engineered to run associated and dedicated Galaxy workflows for the detection and eventually classification of pathogens. The web application allows easy interaction with existing Galaxy metagenomic workflows, facilitates the organization, exploration and aggregation of the most relevant sample-specific sequences among millions of genomic sequences, allowing them to determine their relative abundance, and associate them to the most closely related organism or pathogen.

The user-friendly Django-Based interface, associates the users’ input data and its metadata through a bio-IT provided set of resources (a Galaxy instance, and both sufficient storage and grid computing power). Galaxy is used to handle and analyze the user’s input data from loading, indexing, mapping, assembly and DB-searches. Interaction between our application and Galaxy is ensured by the BioBlend library, which gives API-based access to Galaxy’s main features. Metadata about samples, runs, as well
as the workflow results are stored in the LIMS. For metagenomic classification and exploration purposes, we show, as a proof of concept, that integration of intuitive exploratory tools, like Krona for representation of taxonomic classification, can be achieved very easily. In the trend of Galaxy, the interface enables the sharing of scientific results to fellow team members.

**Keywords**
High Throughput Sequencing, Next-Generation Sequencing, Laboratory Information Management System, Galaxy, Django

---

Corresponding author: Olivia Doppelt-Azeroual (olivia.doppelt@pasteur.fr)

Competing interests: No competing interests were disclosed.

Grant information: Damien Correia and Olivia Doppelt-Azeroual were financed by the "COMMISSARIAT A L'ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES" in the scope of a national anti-terrorism fight NRBC project. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2015 Correia D et al. This is an open access article distributed under the terms of the Creative Commons Attribution Licence, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Data associated with the article are available under the terms of the Creative Commons Zero "No rights reserved" data waiver (CC0 1.0 Public domain dedication).

How to cite this article: Correia D, Doppelt-Azeroual O, Denis JB et al. MetaGenSense: A web application for analysis and visualization of high throughput sequencing metagenomic data [version 1; peer review: 2 approved with reservations, 1 not approved] F1000Research 2015, 4:86 (https://doi.org/10.12688/f1000research.6139.1)

First published: 02 Apr 2015, 4:86 (https://doi.org/10.12688/f1000research.6139.1)
**Introduction**

**Background HTS & metagenomics**

The detection and characterization of emerging infectious agents has been a continuing public health concern. High Throughput Sequencing (HTS) Next-Generation Sequencing (NGS) technologies have proven to be promising for unbiased detections of pathogens in complex biological samples. They are efficient and provide access to comprehensive analyses.

In most large-scale genomic (re)sequencing initiatives involving both sequencing technology, genotyping expertise and computational analyses, the ultimate goal targets analysis of the data in a reference-free context. Depending on sufficient sequencing throughput and availability of reference genomes, raw sequence reads are to be handled by de novo assembly protocols. The choice of the most appropriate assembly algorithm will both depend on the number of sequenced DNA fragments and the genome size of the targeted species. Most well-acknowledged computational bottlenecks for those short-read assemblers concern memory footprints and difficulties in correctly handling repetitive sequences. Assembly very often results in discontinuous sequence contigs and hence insufficient genome coverage. Currently, de novo assembly yields better coverage for small genomes (i.e. bacterial and/or viral species), though assembly of genomes in a metagenomics setup is nowadays considered as complicated and very challenging. Concerning species with no reference in public databases, pre-processing steps are required to increase genome coverage. For example, the use of paired-end sequence data using different insert size libraries is a well-established technique to increase assembly scaffold sizes.

Genotype calling from low coverage data may require extra steps of imputation, filling the gaps that remain due to lack of coverage, and results in more accurate genotypes. Identification of candidate haplotypes and inferring the genotype, by either “phasing” the data to known haplotypes or derivation from external reference panels, allows to better characterize missing genotypes among the individuals.

**Bioinformatics and HTS projects**

Current NGS platforms including Illumina, Ion Torrent/Life Technologies, Pacific Bioscience and Nanopore can generate reads of 100–10,000 bases long allowing better coverage of the genome at lower cost. However, these platforms also generate huge amounts of raw data. For example, the raw data produced by an Illumina HiSeq-2500 platform adds up to 1TB per run. Sequencing reads are recorded as FastQ formatted files along with the corresponding quality score for each nucleotide.

In addition to those sequence files, it has become important to also consider and store associated sample related metadata (collection date, location, etc.). Thus, NGS projects usually represent such a huge amount of relevant sample-specific sequences that efficient data management and visualization resources have become mandatory. The challenges accompanying HTS technologies raise the following issues: (1) How do we best manage the enormous amount of sequencing data? (2) What are the most appropriate choices among the available computational methods and analysis tools? The issue concerning the growing amount of data can be managed through a dedicated Laboratory Information Management System (LIMS), solely to organize and provide perspective regarding the information contained. The question regarding the lack of adapted intertwining among the wide spectrum of available tools, was in part filled by workflow management systems, even though it still requires fairly advanced knowledge of the tools available at hand.

Indeed, today, hundreds of bioinformatics tools are available, each with specific parameters and each available either through GUI or command line. Galaxy, is a scientific workflow management system, which provides means to build multi-step computational data-processing, quality control, and analytic results aggregation, while additionally ensuring analysis reproducibility. In addition to a system for composing pipelines, there is a need for an adapted computational infrastructure capable of doing the processing and data storage in a scalable manner.

MetaGenSense is a managing and analytical bioinformatics framework that is engineered to run dedicated Galaxy workflows for the detection and eventually classification of pathogens. It aims to integrate the capacity for large-scale genomic analysis and technical expertise in sequencing and genotyping technology among project partners. The web application was produced in order to facilitate access to high throughput sequencing analysis tools, acting as an information resource for the project and interacting research partners. This user-friendly interface has been designed to associate bio-IT provider resources (a local Galaxy instance, sufficient storage and grid computing power), with the input data to analyse and its metadata. The use of the available Galaxy tools is automated with MetaGenSense. Galaxy, as a pipeline management software, lets you define workflows and pushes the data through that pipeline. The pipeline manager ensures that all the tools in the pipeline get to run successfully, typically spreading the workload over a computational cluster. MetaGenSense is used at the Pasteur Institute to do the bulk of the data processing for a number of HTS projects, and can be adapted to launch any of the software packages available in the Galaxy workflow designer interface. A dedicated LIMS (postgresql-based) was developed to ensure data coherence. In more detail, the web interface design is based on the Django web framework (http://www.djangoproject.com). Moreover, the communication with Galaxy is ensured by the BiobBlend library which provides a high-level interface for interacting with the Galaxy application, promoting faster interaction, and facilitating reuse and sharing of scripts.

**Software tool - implementation**

**MetaGenSense global description**

MetaGenSense is a bioinformatics application that is geared to ease the scientists’ work in management of NGS project-related data and results. MetaGenSense is built upon three major components, two of which are specific to the project: a dedicated LIMS and a Django-based web user-interface. The third component is Galaxy, which is the bioinformatics workflow management system. In the following paragraphs, we describe the interface’s implementation and discuss how communication between the different parts takes place, in a smooth and user-friendly, managing web-user interface.

**A dedicated LIMS**

A LIMS can be described as a software-based laboratory that offers a set of key features that support modern laboratory operations. Those systems have become mandatory to manage the quantity of
metadata related to both raw data and their analysis results, obtained through bioinformatic tools. In this project, the LIMS is based on a postgresql database. It was designed and structured with expert knowledge from biologists and bioinformaticians with sequencing competence, in order to answer their specific needs ensued by the sample management. Its main feature is that it was also designed to store interesting and worth sharing information obtained by the analysis, as well as the information about the type of workflow that was used to perform the bioinformatics treatment. The database’s schema is available in the Supplementary Figure 1. We provide here an excerpt of the existing tables (divided in three categories): (1) experimental data (LIBRARY_PREPARATION, SAMPLE, TECHNOLOGY, RUN, GEOGRAPHIC_LOCATION, GPS_COORDS), (2) bioinformatic metadata (RAW_DATA, FILE_INFORMATION, WORKFLOW_DATA, RUN_WORKFLOW, WORKFLOW), and (3) user and project data (PROJECT, PROJECT_SUBSCRIBERS, AUTH_USER).

MetaGenSense Django-based web user interface

Django is a high-level Python Web framework. It encourages rapid development and clean, pragmatic design. It is used by many known websites. Moreover, the python language (https://www.python.org) has become a reference for scientific applications.

MetaGenSense is divided in 4 sub-applications which are: 1) user management, 2) lims, 3) workflow_remote, 4) analyse. Each has a specific function, and the task-partitioning has been designed to allow independent evolution of each part according to the user’s needs.

1. **user_management**: manages user authentication. Examples of implementations comprehend communication with an LDAP user authentication database, but it can be used as a user management database.

2. **lims**: ensures the organization and the data partitioning according to the selected project. A project contains sample metadata, and enables to share them only with selected. This part of the application handles sample traceability, an important component of any present-day core resource laboratory.

3. **workflow_remote**: is in charge of the communication with Galaxy. It manages: (a) the instance connection, (b) the user histories (c) the data from galaxy libraries, (d) import of data from a data library to a Galaxy user-history instance. (e) Execution of the selected galaxy workflow. This application handles data storage and links the samples to the selected workflow. In practice, this application could access any of the BioBlend functionalities.

4. **analyse**: deals with the workflow result files. The user can choose to “save” a file in order to share the results with the other users involved in the project. Large data result files can be exported using the Galaxy export functionality or can be downloaded (if the results file can be dealt with through a web browser).

Communication with Galaxy

The following paragraph discusses communication between MetaGenSense and Galaxy. Scientists and data-managers use Galaxy to facilitate bioinformatics analysis. A large number of XML formatted tool-configuration files have already been integrated, facilitating the execution of e.g. a mapping tool like BWA through galaxy instead of executing it using the command line.

For programming purposes and in order to interact with Galaxy using command line, the Galaxy team initially implemented a Galaxy-API (which allowed, for example, retrieval of the user list of a Galaxy instance, to create a library for a specific user, etc.). However, this project was rapidly replaced by a highly dedicated and specific python library called BioBlend. This API gives access to most Galaxy functionalities through scripts and command lines. We prototyped our instances of BioBlend, and validated each task that MetaGenSense was submitting to Galaxy (Figure 1). At the time of development, specific functionalities were not fully ready to use (e.g. the Tools.run_tools function), which made us interact with the BioBlend development team for concomitant finishing and perfection of the tools and accompanying API.

As mentioned earlier, the sub-application workflow_remote from the web interface uses BioBlend functionalities described in Figure 1.

![Architecture Diagram](image-url)

**Figure 1.** Communication details between MetaGenSense and Galaxy using BioBlend.
Data management

Everything is integrated and automated except the big data management. Indeed, MetaGenSense senses when new files are copied within the exchange galaxy project directory, but those data need to be copied there using a UNIX terminal or a FileZilla-like solution (https://filezilla-project.org/).

Pre-designed Galaxy workflows

The MetaGenSense project was initially implemented and validated for metagenomic analyses; most of its uses concern two prototyped workflows designed to preprocess raw fastq data, analyse it and determine the taxonomy distribution within the sample. However, any other type of workflow can be associated to the MetaGenSense application. This only requires an admin user and a workflow identifier.

Case study - use example

We exemplify a use-case of MetaGenSense’s use through the analysis of a batch of biological samples for a dedicated project. The necessary steps to obtain a running MetaGenSense instance, with management of project data and analysis using workflows, are the following:

0/ Log onto MetaGenSense.

1/ Creation of a new project, with a name, a context, a short description and (most importantly) the other persons involved in the project.

2/ Start filling the LIMS database. Enter: a, the sample information, b, the library sequencing protocol, c, the run details and d, the raw data file list. The raw data will be subjected to bioinformatic analysis.

3/ At this step, the user needs to use a terminal (or a FileZilla-like tool) to connect to its transfer directory. Create a subdirectory named after the project, and copy the raw data in that directory. This protocol enables MetaGenSense to detect (Sense) the files that will be copied into the Galaxy instance and analysed.

4/ Back on the MetaGenSense GUI, the user needs to click on the “Workflows” button, and click on “import new files” button to import into Galaxy the inputs that were transferred at the previous step.

5/ Create a Galaxy history,

6/ select the workflow,

7/ select the workflow input(s),

8/ launch the analysis,

9/ follow the workflow status,

10/ At each step, the user has three choices: If the results file sizes are larger than 2 GB, they can be exported (using the native Galaxy tools), or if the file sizes are smaller than 2 GB, they can be downloaded or saved in the LIMS, be tagged as interesting and shared with other project members.

11/ Visualization of the results by clicking on the “Analyse” button. All workflow inputs as well as LIMS result files are visible on this tab. Krona’s representation can easily be visualized if stored in html files.

Discussion and conclusions

The technology evolution in molecular biology, especially in NGS, has moved biology into the big data era (consisting of handling data, computation requirements, efficient workflow design, and knowledge extraction). With this trend, the challenges faced by life scientists have been shifted from data acquisition to data management, processing, and knowledge extraction. While many studies have recognized the big data challenge, few systematically present approaches to tackle it. New findings in biological sciences usually come out of multi-step data pipelines (workflows). Galaxy is such a workflow-managing tool dealing with big data. However, it is still necessary to globally optimize the data flow in an overall multi-step workflow in order to eliminate unnecessary data movement and redundant computation. On the other hand, data information traceability has become an inevitable requirement in a present-day laboratory setup. In the meantime, knowledge-embedded data and workflows are expected to be an integral part of future scientific publications.

We therefore, engineered MetaGenSense, a Django-based web interface which helps biologists, who are unfamiliar with the design of Galaxy workflows, to quickly obtain analysis results from HTS sequencing projects. It uses Galaxy as a workflow management software and the BioBlend API to remotely manage data upload, workflow execution as well as analysis of results. MetaGenSense covers data processing up to presentation of data and results in a genome browser compatible data format. Its main advantages encompass data handling through its incorporated LIMS, user and project handling in a cooperative context, it enables data sharing without compromising data confidentiality, it features automated workflow execution, resulting altogether in decreasing the data and analysis delivery time. MetaGenSense is available as open-source from GitHub, and can be deployed very easily. Though the prototyped tool is mainly focused on metagenomic sample analysis, its modularity allows it to be easily complemented, through project-specific Galaxy workflows, for a variety of other NGS related initiatives.

Software availability

Latest source code
https://github.com/pgp-pasteur-fr/MetaGenSense

Source code as at the time of publication
https://github.com/F1000Research/MetaGenSense

Archived source code as at the time of publication
DOI: 10.5281/zenodo.16510

License: GPLv2

Author contributions
DC, ODA, MV, JBD and VC designed and implemented the software. DC, ODA and MV wrote the manuscript. VC supervised the project.
contributed to discussion and reviewed the manuscript. All authors approved the final manuscript.

Competing interests
No competing interests were disclosed.

Grant information
Damien Correia and Olivia Doppel-Azeroual were financed by the “COMMISSARIAT A L’ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES” in the scope of a national anti-terrorism fight NRBC project.

I confirm that the funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Acknowledgements
Pathoquest, CEA/DSV/IG/Genoscope, Fabien Mareuil (part of the CIB team).

Supplementary materials

Supplemental Figure 1. LIMS database schema.
Open Peer Review

Current Peer Review Status: ? × ?

Version 1

Reviewer Report 24 June 2015

https://doi.org/10.5256/f1000research.6578.r8902

© 2015 Hiltemann S. This is an open access peer review report distributed under the terms of the Creative Commons Attribution Licence, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Saskia Hiltemann
Department of Bioinformatics, Erasmus University Medical Center, Rotterdam, The Netherlands

General Comments:

The authors describe their application, MetaGenSense, a web-based application for analysing metagenomic data. It provides a user-friendly interface which combines a LIMS system with a Galaxy backend for computation and workflow management.

The Django framework is very nice, and I think the integration of Galaxy with a LIMS system is very useful and something many readers will be interested in.

However, many aspects of this application are tailored specifically to the authors' local setup. I installed parts of the application, but since no information is provided on how to install the various components (LIMS/Django/KRONA/BioBlend), and since the Galaxy server used in the code was not accessible to me, it was not fully functional, and because documentation was lacking, it was unclear to me how to proceed with the setup.

To make this work more valuable to the readers, the following additions would be helpful:

1. Installation instructions for the code on GitHub. The readme file is empty at the moment.

   How to install the various components (LIMS, Django UI, KRONA, BioBlend)? And how to connect the different components together? How to configure the webserver correctly (apache/nginx/other)? Which parts of the code are specific to the authors' local setup and need to be adapted when readers install their own MetaGenSense instance?

2. A description of the Galaxy workflows used by the authors would also be very interesting, which tools are used? are they available from the Galaxy tool shed?

3. Either create a demo server with an example project or add screenshots of the application to the manuscript. The UI looks quite nice, show it to the readers.
4. The case study section is very technical, and would be enhanced by showing the use-case in terms of a real biological example, add screenshots of a real-world analysis to the various steps in this section.

Minor Edits:

1. Capitalize the word "Galaxy" throughout.

2. In section "Bioinformatics and HTS projects", BiobBlend --> BioBlend

Competing Interests: No competing interests were disclosed.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 06 Aug 2015

Olivia Doppelt-Azeroual, Institut Pasteur, F-75724, Paris, France

Thank you very much for your critical review. We posted the answers for your remarks underneath them in bold.

To make this work more valuable to the readers, the following additions would be helpful:

1. Installation instructions for the code on GitHub. The readme file is empty at the moment.

   Yes, the README file is now complete. It contains installation information. Please tell us if you feel that it gathers all useful information to install and configure MetaGenSense.

   Along with the version 2 of MetaGenSense publication, we built a release of the software. It is available at the URL (https://github.com/pgp-pasteur-fr/MetaGenSense/releases/tag/v1.0)

   How to install the various components (LIMS, Django UI, KRONA, BioBlend)? And how to connect the different components together? How to configure the webserver correctly (apache/nginx/other)? Which parts of the code are specific to the authors' local setup and need to be adapted when readers install their own MetaGenSense instance?

   Concerning the LIMS, Django, BioBlend, everything is in the application itself so the connection between the components is natively implemented.

   For the apache, it is directly linked to Django which is deployed on an apache server. It is very well documented on this url:

   https://docs.djangoproject.com/en/1.8/howto/deployment/

   For KRONA, the javascript which enables the taxonomy distribution exploration is actually generated by a tool, installed in Galaxy. At the Institut Pasteur, it is part of an in-house package which gathers several tools for taxonomy analyses. Those tools are available on GitHub now (it was not a year ago) at the URL: https://github.com/C3BI-pasteur-fr/taxo_pack. To test those tools, we implemented
a Virtual Machine image pre-configured to test MetaGenSense directly on a web browser. Instructions are also available in the GitHub README file.

For your last question, please look at the “set the settings” part of the README file.

2. A description of the Galaxy workflows used by the authors would also be very interesting, which tools are used? are they available from the Galaxy tool shed?

The workflow included on the virtual Machine Galaxy instance contains a light version of our metagenomic analysis workflow. A small fastq file is also included to test it.

3. Either create a demo server with an example project or add screenshots of the application to the manuscript. The UI looks quite nice, show it to the readers.

Thank you for that. Yes, another reviewer suggested that we put screenshots of the UI. We added a figure that resumes all the windows and their use for each step. The figure is available at this url: https://github.com/pgp-pasteur-fr/MetaGenSense/blob/master/doc/images/metagensense.png as well as each of the steps (in bigger picture). This figure is now added to the publication.

4. The case study section is very technical, and would be enhanced by showing the use-case in terms of a real biological example, add screenshots of a real-world analysis to the various steps in this section.

Please tell us if the figure resumes well the various steps.

For the minor remarks, they are all included in the second version of the publication. Thank you for your critical review of our article.

**Competing Interests:** No competing interests were disclosed.
Laboratory Information Management System (LIMS). The application can fetch that information and pipe it into predefined Galaxy workflows, run them and visualise the output via a framework called KRONA.

The introduction to the article is perhaps too long (almost half of the article). There are sections that are not necessarily related to the research presented here, e.g., the paragraph focusing on the assembly problem of next generation sequencing reads. It would be useful, however, that the authors give a more comprehensive introduction into metagenomics as this topic is only covered very briefly at the beginning of the introduction.

The section on the software tool itself is very technical. I have trouble identifying a clear train of thought. Also it could be shorter and more precise. The case study does not really seem to be a case study on how the application can be used to actually find pathogen information in metagenomic data but is more like a step by step protocol on how to use the application. I suggest that this kind of information should be moved to the documentation and that instead a concrete biological example is demonstrated in the article. Moreover, the title says that MetaGenSense can visualise its results. However, this is not shown in the article. Therefore I would advise the authors to consider replacing the current figure with a figure demonstrating the results of a concrete biological use case.

The discussion and conclusion seem to be a summary rather than a discussion. MetaGenSense seems to lack many of the standard requirements of a quality software product

- We could not find any documentation. The README file on GitHub does not contain any information.
- The last update to the code was months ago, suggesting that the program is not being developed and maintained actively.
- We could not find any tests.
- We could not find any examples, demos or even screenshots of the interface.

Therefore we are not convinced that MetaGenSense adheres to the journal’s quality standards. We believe that the article should be revisited and documentation, live examples and tests should be added to the software before the article should be indexed.

**Competing Interests:** No competing interests were disclosed.

*We have read this submission. We believe that we have an appropriate level of expertise to state that we do not consider it to be of an acceptable scientific standard, for reasons outlined above.*

Author Response 28 Aug 2015

Olivia Doppelt-Azeroual, Institut Pasteur, F-75724, Paris, France

We tried to answer each of your comments. A second version of the publication will be available soon.

Thank you for your critical review of our article.

Reviewer Comment:
MetaGenSense is intended to help find pathogen data in metagenomic data created through next generation sequencing. Measured data including the sequencing reads and metadata are fed into a Laboratory Information Management System (LIMS). The application can fetch that information and pipe it into predefined Galaxy workflows, run them and visualise the output via a framework called KRONA.

The introduction to the article is perhaps too long (almost half of the article). There are sections that are not necessarily related to the research presented here, e.g., the paragraph focusing on the assembly problem of next generation sequencing reads. It would be useful, however, that the authors give a more comprehensive introduction into metagenomics as this topic is only covered very briefly at the beginning of the introduction.

Introduction is now shorter in the second version of the publication. It was modified in a way that more emphasis is made on the importance of a true metagenomics component for HTS-type analyses. Information regarding assembly problems are indeed of secondary importance, and they were removed for the new version.

Reviewer Comment:

The section on the software tool itself is very technical. I have trouble identifying a clear train of thought. Also it could be shorter and more precise. The case study does not really seem to be a case study on how the application can be used to actually find pathogen information in metagenomic data but is more like a step by step protocol on how to use the application. I suggest that this kind of information should be moved to the documentation and that instead a concrete biological example is demonstrated in the article. Moreover, the title says that MetaGenSense can visualise its results. However, this is not shown in the article. Therefore I would advise the authors to consider replacing the current figure with a figure demonstrating the results of a concrete biological use case.

Thank you for this interesting remark; the use-case part of the article was modified to be more precise. Moreover, a user can now use MetaGenSense on a concrete biological dataset. We created a virtual machine image preconfigured to directly use MetaGenSense with a web browser. A fastq file, as well as a light version of our metagenomic Galaxy workflow is included. The user can connect to the framework, launch an analysis, explore the framework, as well as the results.

Moreover, we also would like to change the title of the publication for it to be more adapted to our approach; replacing visualisation by the word exploration which is really the goal of an application like MetaGenSense. The title of the second version of the article is:

"MetaGenSense: A web application for analysis and exploration of high throughput sequencing metagenomic data."

Reviewer Comment:

The discussion and conclusion seem to be a summary rather than a discussion.

MetaGenSense seems to lack many of the standard requirements of a quality software product

- We could not find any documentation. The README file on GitHub does not contain any information.
MetaGenSense README file is now complete. We also have written on a userGuide available directly on our Github repository through the web tool readthedocs: http://metagensense.readthedocs.io.

**Reviewer Comment:**
- The last update to the code was months ago, suggesting that the program is not being developed and maintained actively.

The code in the GitHub repository was committed just before the submission of the article. A few debug and add-ons have been implemented since the previous release.

**Reviewer Comment:**
- We could not find any tests.

As mentioned earlier, we implemented a Virtual Machine Image containing the infrastructure to test our framework. It is pre-configured so that any user can start using MetaGenSense with a web browser. It is available on the Institut Pasteur server as it was too big to be uploaded on GitHub (http://webext.pasteur.fr/metagensense/metagensense.ova). Documentation about this image is available on the README file of our GitHub repository. As Metagenomic analyses are time and storage consuming, we made available a very light version of our workflow with a small fastq file. However, it is enough to test the framework and to understand how the database, the Django framework and the related Galaxy instance are working together.

**Reviewer Comment:**
- We could not find any examples, demos or even screenshots of the interface.

For the screenshots, we added a picture of the framework at each step of its use. It is also available at the url: https://github.com/pgp-pasteur-fr/MetaGenSense/blob/master/doc/images/metagensense_complete.jpeg

Moreover, as the screenshots are small, they are also available on our github repository in the doc directory.

**Competing Interests:** No competing interests were disclosed.

Reviewer Report 19 June 2015

https://doi.org/10.5256/f1000research.6578.r8901
General comments

The authors have developed MetaGenSense, a web-based application for analysing metagenomics data. MetaGenSense also acts as a laboratory information management system, managing the metagenomics data and the results of their analysis by Galaxy workflows.

The work done by the authors sounds promising since I am a fan of Galaxy and of the Django framework. However, I believe the paper can be much improved by providing additional details in a number of places.

For example, it would be interesting to learn more about the two prototyped workflows for analysing metagenomics data which are alluded to in the “Pre-designed Galaxy workflow” section on page 5.

The “Case study - use example” section could also be improved by providing screenshots of the MetaGenSense GUI which are relevant to each or some of the steps. At the moment, I have no idea what the GUI for MetaGenSense looks like since there is also no example instance of MetaGenSense available on the Web which would have been useful for reviewing this paper.

In the final paragraph, it is stated that MetaGenSense can be easily deployed but looking at the source code in https://github.com/F1000Research/MetaGenSense, there appears to be a lack of documentation to enable me to do this. For example, which version of Django should I use with MetaGenSense? How do I get the MetaGenSense source code integrated with the Django framework? The LIMS for MetaGenSense uses PostgreSQL; how do I set up this database and get it linked to MetaGenSense? How do I provide MetaGenSense with access to a Galaxy server? I think the authors need to provide this type of information to help readers install MetaGenSense for the authors' source code to be more useful.

Minor corrections

Page 3:

Paragraph 4
Pacific Bioscience should be Pacific Biosciences.

Paragraph 7
BiobBlend should be BioBlend.

Page 4
workflow_remote should be workflow_remote
like BWA through galaxy - should be, "like BWA through Galaxy..."

Case study - use example: Need to be consistent by starting bullet points with capital letters.

**Competing Interests:** No competing interests were disclosed.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 06 Aug 2015

**Olivia Doppelt-Azeroual**, Institut Pasteur, F-75724, Paris, France

Thank you for your review. For each of your remarks, we wrote responses in bold:

For example, it would be interesting to learn more about the two prototyped workflows for analysing metagenomics data which are alluded to in the "Pre-designed Galaxy workflow" section on page 5.

This publication aims to present an application combining as you wrote above a lims, a direct link to any Galaxy and a way to sort and manage Galaxy results. The workflow you choose to use is totally arbitrary. In essence, MGS was designed in a way that any Galaxy workflow can be plugged-in.

To facilitate testing, we implemented a Virtual Machine image pre-configured to test MetaGenSense directly on a web browser. Instructions for download and use are available in the GitHub README file. However, as instructions for installation of the framework are now available, any developer can download and link MetaGenSense to his Galaxy.

The "Case study - use example" section could also be improved by providing screenshots of the MetaGenSense GUI which are relevant to each or some of the steps. At the moment, I have no idea what the GUI for MetaGenSense looks like since there is also no example instance of MetaGenSense available on the Web which would have been useful for reviewing this paper.

Yes we agree with you, we omitted to add screenshots in the first version of the article. We added a new figure that resumes all MetaGenSense steps and functionalities. It is available at: https://github.com/pgp-pasteur-fr/MetaGenSense/blob/master/doc/images/metagensense_complete.jpeg. It is now the 2nd figure of the article. Moreover all the small figures (steps) are also available in the doc directory of our GitHub repository.

In the final paragraph, it is stated that MetaGenSense can be easily deployed but looking at the source code in https://github.com/F1000Research/MetaGenSense, there appears to be a lack of documentation to enable me to do this. For example, which version of Django should I use with MetaGenSense? How do I get the MetaGenSense source code integrated with the Django...
framework? The LIMS for MetaGenSense uses PostgreSQL; how do I set up this database and get it linked to MetaGenSense? How do I provide MetaGenSense with access to a Galaxy server? I think the authors need to provide this type of information to help readers install MetaGenSense for the authors' source code to be more useful.

We improved the documentation on our github repository (https://github.com/pgp-pasteur-fr/MetaGenSense) with elements facilitating deployment. We completed a README file containing well-informed installation and configuration instructions. We also wrote a UserGuide working with readTheDocs documentation tool (http://metagensense.readthedocs.io). Don’t hesitate to tell us if parts of the documentation are still a bit blurry.

We will adapt the manuscript with all your minor corrections and suggestions.

Thank you very much for your review.

Competing Interests: No competing interests were disclosed.

Reviewer Response 27 Sep 2016

Peter Li, GigaScience, Hong Kong SAR, China

To test the authors' software, it will take over 6 hours for me to download the 2.7 GB machine image provided by the authors. I think this is due to my office internet connection being very slow and because of the fact that I am based in the Far East. Unfortunately, waiting 6 hours plus for the machine image to download is not really practical for me but this is not a fault of the authors.

I had a look at the instructions to install MetaGenSense which are available from the README.md file in their GitHub repository in an attempt to manually install the software. I feel that the instructions are minimal, for example, the installation requires a requirements.txt file but I could not find one in the MetaGenSense repo. You would need to know that the contents of requirements.txt are listed at the start of the manual installation instructions. The installation instructions also rely on the user doing background reading to find out how to install and use MetaGenSense's dependencies, for example, virtualenv and a database server.

Competing Interests: No competing interests were disclosed.

Author Response 21 Nov 2016

Olivia Doppelt-Azeroual, Institut Pasteur, F-75724, Paris, France

Dear Peter Li,

Thank you for your review and remarks.

Since last August, we asked a non familiar with MetaGenSense colleague to read and install locally the application. This exercise has enabled us to clarify some points. We simplified the README file (https://github.com/pgp-pasteur-fr/MetaGenSense) and added significant details to the installation procedure part in the readthedocs documentation:

- As you mentioned, the virtualenv installation procedure has been added.
We also clarified the part concerning the requirements, i.e. the developer needs to copy the three requirements in a requirements.txt file and use the "pip" command to install them. http://metagensense.readthedocs.io/en/latest/installation.html#requirements

The part concerning the database configuration has also been a bit modified in order to be clearer.

We hope the changes applied recently enable you to successfully install MetaGenSense.

Best regards,

Olivia Doppelt-Azeroual

**Competing Interests:** None